

A program a **PRIMO** nagyfelbontású grafikájának támogatására készült. A jelenlegi változat 68 új utasítással bővíti ki az eredeti BASIC-et, valamint a **LIST**, **EDIT** és **DELETE** utasítások hibáit is kijavítja. Az összes új utasítás használható mind parancs, mind program módban.

A **LIST**, **EDIT** és **DELETE** megszüntetett hibái:

=====

1. Ha egy sorban macskakörmök közé / a lenyil használatával / 128 feletti kódu karaktereket vittünk be, listázáskor, és editáláskor ezek helyett az ilyen kódu utasítások jelennek meg.
2. Ha 0 sorszámú sort javítottunk, a sorszám nem jelent meg.
3. Ha editálás közben **BRK**-et nyomtunk, az editálásból kilépünk ugyan, de soremelés nem történt a képernyőn.
4. Ha 210 karakternél hosszabb sort javítottunk, a gép nem korlátozta a sorhosszt, és megőrült, jobb esetben elszállt. Most ha 210 karakternél hosszabb sort javítunk, a gép addig nem enged beszúrást végezni, míg a sorhosszt 210 karakter alá nem csökkentjük.
5. Ha **DELETE** utasításban nemlétező sorszámmra hivatkoztunk, vagy **DELETE nn**-utasítást próbáltunk meg használni, hibaüzenetet kaptunk. Ezentúl a **DELETE** utasítás szintaktikája megegyezik a **LIST** utasításával, egyetlen különbség, hogy ha csak **DELETE**-et írunk akkor nem törlődik az egész program, hanem hibaüzenetet kapunk.

A **GRAPHICS** új utasításai:

=====

Képernyőtár - helyzet kezelő utasítások: / paramétere ezek közül egyik utasításnak sincs /

LLOC : Második képernyőtárat hoz létre az eredetinel 8 Kbyte-al alacsonyabb címre, és a BASIC végét erre a címre állítja, majd egy **CLEAR 50** parancsot hajt végre. / tehát törli a változókat /. Célszerű, ha ez a programunk első utasítása, ha két képernyőre van szükségünk. Csak 16 Kbyte RAM-nál nagyobb memóriájú gépekben működik.

SWITCH : Csak már létrehozott második ernyőtár esetén működik. Átkapcsolja a képernyőtár címét a **SOFTWARE**-ben és a **HARDWARE**-ben a képrajzolás végén. A funkció oda-vissza kapcsol a két képernyő között.

SCHANGE : Csak már létrehozott második képernyőtár esetén működik. Átkapcsolja a **SOFTWARE**-ben a képernyőtár címét. Így lehetővé válik az, hogy a gép más képernyőn rajzol, mint ami látszik a TV-n. A funkció oda vissza kapcsol.

HCHANGE : Csak már létrehozott második képernyőtár esetén működik. Átkapcsolja a **HARDWARE**-t a másik képernyőtárra. Célja hasonló a **SCHANGE** céljához, csak ennél a funkciónál váltás történik a képernyőn: A funkció oda-vissza kapcsol.

ERASE : A feleslegessé vált második képernyőtárat törli, visszakapcsol a normális képernyőtárra, a BASIC végcímét visszaállítja az eredetire, és egy **CLEAR 50** parancsot ad ki.

2. Mozgató utasítások:

A mozgató utasításokkal a képernyő kijelölt téglalap alakú területeit lehet különböző irányban, különböző típusú finomscrolllokkal elmozgatni. A mozgató utasításokon kívül ide sorolható még két utasítás, amelyek a mozgató utasítások használatát könnyítik meg. Ezek:

PAR : paraméterek: A használni kívánt mozgató utasítás száma és ennek paraméterei.
Ez az utasítás lehetővé teszi, hogy a programunk paraméteresen állítsa be, hogy mely mozgató utasítást használja.

REPEAT : paraméter : DARABSZÁM.
Az utoljára végrehajtott mozgató utasítást ismétli az előírt darabszámúszor.

Most következik a tényleges mozgató utasítások leírása. A mozgató utasítások elnevezése nem véletlenszerű, hanem bizonyos logika szerint történt. Éppen ezért ezek az elnevezések elég furcsák. Éppenhol a logika megtörik, mert az éppem hiányzó utasításnak nem volt különösebb értelme.

Az egyes betűk jelentése:

Az első betű a mozgató irányát határozza meg.
J=jobbra, B=balra, F=fel, L=le, V=vegyes /az első sort jobbra, a másodikat balra, a harmadikat megint jobbra és így tovább /
Az utolsó betű a mozgató típusát határozza meg.
R=roll : a mozgató irányába eső oldalon kimenő pontok a szemközti oldalon visszajönnek.
S=scroll: a mozgató irányába eső oldalon kimenő pontok helyett a szemközti oldalon sötét vagy világos pontok jönnek be.
A közbeeső betűk száma lehet nulla, ezek az alapesetek, illetve lehet egy, kettő vagy három.
Ha csak egy betű van az irány és típus között, akkor az vagy I, vagy B. Az I betű jelentése : inverz. Ez azt jelenti, hogy azok a pontok melyek kimentek, invertálva jönnek vissza, illetve scroll esetén a sötét pontok helyett világosak jönnek be. A B betű jelentése : Byte-on belül. Ez azt jelenti, hogy a terület Byte-jainak határán nem tolódnak át a pontok, hanem a Byte-on belül kerülnek mozgásra.
Ha két betű szerepel az irány és a típus között, akkor az vagy IB, vagy BI. Az IB betűpár az I és a B funkció keveréké jelenti, azaz byte-okon belüli inverz mozgatót határoz meg. A BI betűpár azt jelenti, hogy a byte határon invertáló mozgató. Ez azt jelenti, hogy a terület byte-határain a pontok áttolódnak /tehát nem byte-on belüli mozgató/, de közben meginvertálódnak.
Ha három betű szerepel az irány és a típus között, akkor az csak IBI lehet. Jelentése : Inverz, Byte határon invertáló mozgató. Ez a funkció a byte határokon áttolódó pontokat is meginvertálja, és a terület határain kimenő pontokat is meginvertálja.

Tipusra nézve a mozgató utasítások közé sorolható még három olyan utasítás, mely tulajdonképpen nem az, mert mozgatót nem végez a terület pontjain. Ezek a **BSET**, a **BRESET**, és a **BINV**. A **BSET** kigyújtja a terület pontjait, a **BRESET** eloltja, és a **BINV** meginvertálja azokat.

Létezik még egy speciális scroll, mely a CLR utasítással hívható. Ez a speciális scroll a terület byte-jait jobbra, majd balra, majd megint jobbra, stb. mozgatja, amíg a területnek nincs vége, majd ezt a következő pontsorban megismétli. Ezzel az utóbbi négy utasítással együtt 42 mozgató utasítás állítható össze. A következő táblázatban felsoroljuk az összes mozgató utasítást, a számukkal együtt /melyekkel lehet rájuk hivatkozni a PAR utasításban/. A REPEAT utasítás kódja a PAR utasításban a 0. / az egyszerűség kedvéért a byte-on belüli mozgásokat byte mozgásnak nevezzük /

A 42 mozgató utasítás :

1.	JR	= Jobbra Roll
2.	JS	= Jobbra Scroll
3.	BR	= Balra Roll
4.	BS	= Balra Scroll
5.	JIR	= Jobbra Inverz Roll
6.	JIS	= Jobbra Inverz Scroll
7.	BIR	= Balra Inverz Roll
8.	BIS	= Balra Inverz Scroll
9.	JBR	= Jobbra Byte Roll
10.	JBS	= Jobbra Byte Scroll
11.	BBR	= Balra Byte Roll
12.	BBS	= Balra Byte Scroll
13.	JIBR	= Jobbra Inverz Byte Roll
14.	JIBS	= Jobbra Inverz Byte Scroll
15.	BIBR	= Balra Inverz Byte Roll
16.	BIBS	= Balra Inverz Byte Scroll
17.	FR	= Fel Roll
18.	FS	= Fel Scroll
19.	LR	= Le Roll
20.	LS	= Le Scroll
21.	FIR	= Fel Inverz Roll
22.	FIS	= Fel Inverz Scroll
23.	LIR	= Le Inverz Roll
24.	LIS	= Le Inverz Scroll
25.	JBIR	= Jobbra Bytehatáron Invertáló Roll
26.	JIBIR	= Jobbra Inverz Bytehatáron Invertáló Roll
27.	BBIR	= Balra Bytehatáron Invertáló Roll
28.	BIBIR	= Balra Inverz Bytehatáron Invertáló Roll
29.	BSET	= /Block SET/ kigyújtja a terület pontjait
30.	BRESET	= /Block RESET/ eloltja a terület pontjait
31.	BINV	= /Block INVERT/ meginvertálja a terület pontjait
32.	CLR	= /CLEAR/ 8 lépésben kitörli a terület pontjait
33.	VR	= Vegyes Roll
34.	VS	= Vegyes Scroll
35.	VIR	= Vegyes Inverz Roll
36.	VIS	= Vegyes Inverz Scroll
37.	VBR	= Vegyes Byte Roll
38.	VBS	= Vegyes Byte Scroll
39.	VIBR	= Vegyes Inverz Byte Roll •
40.	VIBS	= Vegyes Inverz Byte Scroll
41.	VBIR	= Vegyes Bytehatáron Invertáló Roll •
42.	VIBIR	= Vegyes Inverz Bytehatáron Invertáló Roll

8 lépésben volt

A mozgó utasítások paramétere i :

Ha egy mozgó utasításnak csak egy paramétere van, akkor ez azt a számot jelenti, amennyiszer azt a mozgó utasítást meg kell ismételni az aktuális területen.

Ha van második paraméter is, akkor kell, hogy legyen még egy harmadik és egy negyedik is. Ezek rendre $X1, Y1, X2, Y2$ koordináták, melyek közül $X1$ -et és $X2$ -öt a gép úgy kerekíti, hogy pont byte határra mutassanak. A kisebbet lefelé, a nagyobbát felfelé kerekíti, és az így nyert koordinátákkal jellemezhető átlóju téglalap lesz az aktuális terület.

Ha van ötödik paraméter, akkor az a végrehajtás darabszámát jelenti /értéke 1-től 32767-ig terjedhet, ill. -32767-től -1-ig, de negatív szám esetén 65536-ot hozzá kell adni a megadott értékhez, hogy a tényleges darabszámot megkapjuk/

Grafikai utasítások :

A program ismer 11 normális grafikai utasítást. Ezek funkciói, paramétereik leírása található itt. Ezekre az utasításokra nem lehet hivatkozni a PAR utasításban, mivel ennek nem sok értelme lenne.

A 11 normális grafikai utasítás :

LINE $X1, Y1, X2, Y2, \text{Tipus}$

Összeköti egy megadott típusú egyenessel az $X1, Y1$ koordinátájú pontot az $X2, Y2$ koordinátájú ponttal. / LINE="egyenes" /

REC $X1, Y1, X2, Y2, \text{Tipus}$

Az $X1, Y1, X2, Y2$ átlóval rendelkező téglalap megrajzolja a megadott tipussal. /REC tangle="téglalap"/

BLOCK $X1, Y1, X2, Y2, \text{Tipus}$

Az $X1, Y1, X2, Y2$ átlóval rendelkező téglalap egész területét beszínezi a megadott tipussal. /BLOCK="tömb"/

ELLIPSE $X, Y, XR, YR, \text{Tipus}$

Az X, Y középponttal rendelkező, vízszintesen XR , függőlegesen YR sugarú ellipszist pontonként kiszámítja, és megrajzolja a megadott tipussal. /ELLIPSE="ellipszis ; A funkció kissé lassú, de nagyon pontos ellipszist ad, ezért született az EGG funkció, ami jóval gyorsabb, de nem ad olyan pontos eredményt, mint az ELLIPSE/

EGG $X, Y, XR, YR, \text{Tipus}$

Az X, Y középponttal rendelkező, vízszintesen XR , függőlegesen YR sugarú ellipszist megrajzolja a megadott tipussal. /EGG="tojás"/

POLYGON $X, Y, XR, YR, \text{Tipus}, \text{elforgatás}, \text{csúcsok száma}$

Az X, Y középponttal rendelkező, vízszintesen XR , függőlegesen YR sugarú ellipsziszbe rajzolható megadott csúcsszámú sokszöget megrajzolja a megadott elforgatással, és a megadott tipussal. /az elforgatást radiánban kell megadni !/ Nulla elforgatás esetén az első csúcspontot az $X+XR, Y$ koordinátájú pontból kezdi el rajzolni. /POLYGON="sokszög"/

DIAG $X, Y, XR, YR, \text{tipus}, \text{elforgatás}, \text{csúcsok száma}$

Az X, Y középponttal rendelkező, vízszintesen XR , függőlegesen YR sugarú ellipsziszbe rajzolható, megadott csúcsszámú sokszög átlóit megrajzolja a megadott elforgatással, és megadott tipussal. /Az elforgatást radiánban kell megadni ! Nulla elforgatásnál az első csúcspontot az $X+XR, Y$ koordinátájú pontból kezdi el rajzolni. /DIAGonals="átlók"/

5
ARC X,Y,XR,YR,Tipus, kezdőszög, végszög

Az X,Y középponttal rendelkező, vízszintesen XR, függőlegesen YR sugarú ellipszis kezdőszögtől végszögig terjedő részét megrajzolja a megadott tipussal./ a két szöget radiánban kell megadni !; ARC="iv"/

TEXT X,Y,XN,YN,Tipus,"szöveg"

Az X,Y bal ~~széle~~ koordinátából vízszintesen XN, függőlegesen YN nagyítással, és a megadott tipussal kirajzolja a szöveget.
/TEXT="szöveg"/

PAINT X,Y,Tipus /ebben az esetben a tipus csak 1, vagy 0 lehet !/
Azt a zárt alakzatot, mely magába foglalja az X,Y koordinátájú pontot beszínezi. Az alakzatnak nem kell konvexnek lennie. Ha az alakzat nem zárt, akkor a képernyő végével a program automatikusan zárt alakzatnak tekinti azt. Ha a tipus 1, akkor az alakzat határának ki-gyújtott pontokat vesz a program, ha nulla, akkor eloltott pontokat.
/PAINT="fest"/

INV /X,Y/

Az X,Y koordinátákkal rendelkező pontot meginvertálja.

/ A következő műveletsort helyettesíti :

IF POINT /X,Y/ THEN RESET/X,Y/ ELSE SET/X,Y/

A típus jelentése :

Ha a típus nulla, akkor az aktuális pontokat eloltja, ha egy, akkor kioltja azokat, ha kettő, akkor meginvertálja a pontokat.

2